# Install OpenCV4 on ubuntu 18.04 for C++ and Python3

## 相关资源链接

- OpenCV官方相关链接：
    - OpenCV tutorial
    - Face Recognition with OpenCV
    - OpenCV GitHub: 源码以及可离线下载的3rd_party包；
- 百度网盘统一下载-(密码: `304r`）：
    - ORL数据集-The Database of Faces
    - opencv-4.0.1.zip(100.7MB)；
    - opencv_contrib-4.0.1.zip(61.9MB)；
    - ippicv_20109...20180823.tgz(31MB): 密码： `1miu` ；
    - face_landmark_model.dat(69MB): 密码: `fkf0`；

## 下载 & 安装

```
# 若网速即使kb/s，可直接从百度网盘统一下载~
$ git clone https://github.com/opencv/opencv.git
$ git clone https://github.com/opencv/opencv_contrib.git
```

```
$ lsb_release -a; getconf LONG_BIT
No LSB modules are available.
Distributor ID: Ubuntu
Description:  Ubuntu 18.04.2 LTS
Release:  18.04
Codename: bionic
64
$ sudo apt install -y build-essential cmake git pkg-config libgtk2.0-dev
libgtk-3-dev libopenexr-dev gfortran libblas-dev liblapack-dev libeigen3-dev
python-dev python3-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev
libtiff-dev libdc1394-22-dev libjasper-dev libgstreamer1.0-dev libgstreamer-
plugins-base1.0-dev libavcodec-dev libavformat-dev libswscale-dev libavutil-
dev libavresample-dev libxvidcore-dev libx264-dev libv4l-dev

$ unzip opencv-4.0.1.zip && mv opencv-4.0.1 opencv
$ unzip opencv-contrib-4.0.1.zip $$ mv opencv-contrib-4.0.1 opencv-contrib

(torch) $ which python
/home/hxer/miniconda3/envs/torch/bin/python (python3.6)
(torch) $ export ENV_PATH=~/miniconda3/envs/torch  # torch is the env name
# $ENV_PATH is specific with your conda environment and your env name!
(torch) $ cd opencv/build
opencv/build$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
  -D CMAKE_INSTALL_PREFIX=/usr/local \
  -D PYTHON_EXECUTABLE=$ENV_PATH/bin/python3.6 \
  -D PYTHON_INCLUDE=$ENV_PATH/include \
  -D PYTHON_LIBRARY=$ENV_PATH/lib/libpython3.6m.so \
  -D PYTHON_PACKAGES_PATH=$ENV_PATH/lib/python3.6/site-packages \
  -D PYTHON_NUMPY_INCLUDE_DIR=$ENV_PATH/lib/python3.6/site-
packages/numpy/core/include \
  -D BUILD_opencv_python3=ON \
  -D BUILD_opencv_python2=OFF \
  -D INSTALL_C_EXAMPLES=ON \
  -D INSTALL_PYTHON_EXAMPLES=ON \
  -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
  -D OPENCV_ENABLE_NONFREE=ON \
  -D BUILD_EXAMPLES=ON ..  # .. -> $WHERE_OPENCV
  ...skiped...
  -- IPPICV: Download: ippicv_2019_lnx_intel64_general_20180723.tgz
  # [下载时假死]
```

`opencv/3rdparty/ippicv.cmake:47` 处定义了ippicv_xxx.tgz的下载地址：github-opencv/3rdparty/ippicv，由于下载没有进度条、网速很慢，导致用户无法评估当前进度；

从上面给出的百度网盘下载好后，将`tgz`文件拷贝到`.cache/ippicv`下**覆盖**git clone的**同名**文件，由于下载前程序会先查看该`ippicv`文件与github上文件前缀哈希值是否一致(即该文件名前的哈希值)，相等即不再下载，直接使用`.cache`相应文件夹下的文件；实现**离线**下载的目的；

```
$ ls opencv/.cache/ippicv
c0bd78adb4156bbf552c1dfe90599607-ippicv_2019_lnx_intel64_general_20180723.tgz
$ cp ~/Downloads/ippicv_2019_lnx_intel64_general_20180723.tgz
opencv/.cache/ippicv/c0bd78adb4156bbf552c1dfe90599607-
ippicv_2019_lnx_intel64_general_20180723.tgz
```

> Note: 如果注释掉`opencv/3rdparty/ippicv.cmake:47`的github地址，替换成file:///home/.../ippicv_2019...tgz；重新`cmake -D`编译时会出现：
>
> -- IPPICV: Download: ippicv_2019_lnx_intel64_general_20180723.tgz CMake Warning at cmake/OpenCVDownload.cmake:193 (message):
>
> IPPICV: Download failed: 37;"Couldn't read a file:// file"

ippicv离线下载参考：OpenCV 4.0.0编译并通过inter引擎优化流程

离线安装`ippicv`之后继续`cmake -D …`编译出现：

```
-- data: Download: face_landmark_model.dat
CMake Warning at /home/hxer/Code/opencv/cmake/OpenCVDownload.cmake:193
(message):
  data: Download failed: 28;"Timeout was reached"
```

可从GitHub·face_landmark_model.dat或上面给出的百度网盘下载`data/face_landmark_model.dat`文件；

> 下载时 — [data] 就表示该文件下载时的路径为`opencv/.cache/[data/ippicv]/`，下载后和`ippicv`一样**覆盖**到该目录下**同名**文件即可；

```
$ nproc
12
$ make -j12 # 12 CPU cores, 即 nproc输出的当前硬件配置;
$ sudo make install
```

### 配置信息

```
$ sudo echo "/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf
$ sudo ldconfig # 生效
$ echo "export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig/" >>
~/.bashrc
$ source ~/.bashrc
$ ls /usr/local/lib/pkgconfig/
opencv4.pc
$ pkg-config --modversion opencv4
4.0.1
$ pkg-config --modversion opencv
Package opencv was not found in the pkg-config search path.
Perhaps you should add the directory containing `opencv.pc'
to the PKG_CONFIG_PATH environment variable
No package 'opencv' found
```

> - `/usr/local/lib`：shared libraries，都是`libopencv_filename.so.4.0.1`；
> - `/usr/local/include/opencv4/opencv2`为头文件导入目录；
>
> `pkg-config`目录在`$PKG_CONFIG_PATH`环境变量所包含的路径下查找与传入参数`opencv`**同名**的文件`opencv.pc`，从中找出响应的config信息(modvesion).所以在openCV4中需要查询opencv4而不是opencv.

### test for C++

```
$ cd opencv/samples/cpp/exmaple_cmake
$ cmake .  # 输出opencv4.pc中的配置信息
-- OpenCV library status:
--     config: /usr/local/lib/cmake/opencv4
--     version: 4.0.1
--     libraries:
opencv_calib3d;opencv_core;opencv_dnn;opencv_features2d;opencv_flann;opencv_gapi;opencv_highgui;opencv_imgcodecs;opencv_imgproc;opencv_ml;opencv_objdetect;opencv_photo;opencv_stitching;opencv_video;opencv_videoio;opencv_aruco;opencv_bgsegm;opencv_bioinspired;opencv_ccalib;opencv_datasets;opencv_dnn_objdetect;opencv_dpm;opencv_face;opencv_freetype;opencv_fuzzy;opencv_hfs;opencv_img_hash;opencv_line_descriptor;opencv_optflow;opencv_phase_unwrapping;opencv_plot;opencv_reg;opencv_rgbd;opencv_saliency;opencv_shape;opencv_stereo;opencv_structured_light;opencv_superres;opencv_surface_matching;opencv_text;opencv_tracking;opencv_videostab;opencv_xfeatures2d;opencv_ximgproc;opencv_xobjdetect;opencv_xphoto
--     include path: /usr/local/include/opencv4
-- Configuring done
-- Generating done
-- Build files have been written to:
/home/hxer/Code/opencv/samples/cpp/example_cmake
$ make
[100%] Built target opencv_example
$ ./opencv_example # 弹窗title为Sample, 显示非常漂亮 Hello OpenCV 字体;
Built with OpenCV 4.0.1

(opencv_example:12723): GStreamer-CRITICAL **: 19.12.44.090:
gst_element_get_state: assertion 'GST_IS_ELEMET (element)' failed
VIDEOIO ERROR: V4L: can't open camera by index 0
No capture
```

由于显示屏没有摄像头...所以opencv尝试打开摄像头时报错`VIDEOIO ERROR`;

参考：[[笔记] Ubuntu 18.04源码编译安装OpenCV 4.0流程]

## test for Python3

插曲：最开始参考的资料是面向python**2**安装的OpenCV4(都opencv4了，竟然还有人用python2来写)，`apt install`安装的是`python2-dev`、`libgtk2.0-dev`.然后`cmake -D`编译时没有指定conda环境中python相关信息，如：`PYTHON_EXECUTABLE, PYTHON_INCLUDE,PYTHON_LIBRARY, PYTHON_PACKAGES_PATH, PYTHON_NUMPY_INCLUDE_DIR`.

`cmake -D`最初输出匹配pythoninterp的信息：系统python(2.7.9+) 匹配成功，而torch conda 环境下版本为`3.6.2`的python和exactly `3.6.15`不相符，build for python2.

`make install`之后，`python2`可导入`cv2`.但`torch`环境下无法导入；

发现问题后查找资料，在`cmake -D`编译指令中加入`python3`相关命令，同时还关掉了`PYTHON2`的输出，重新编译，输出的总结信息：

```
--   Python 3:
--     Interpreter:
/home/hxer/miniconda3/envs/torch/bin/python3 (ver 3.6.2)
--     Libraries:                   /usr/lib/x86_64-linux-gnu/libpython3.6m.so
(ver 3.6.2)
--     numpy:                       /home/tamim/anaconda3/lib/python3.6/site-
packages/numpy/core/include (ver 1.16.1)
--     packages path:               lib/python3.6/site-packages
--
--   Python (for build):            /usr/bin/python2.7
```

气啊！为何还是`Python (for build)`.修改`PYTHON3_…`回`PYTHON_...`、查看路径是否准确，重新编译了很多次，都是这个结果；有点沮丧；直到看到有人提到`build/CMakeCache.txt`。我将`CMakeCache.txt` 移动到`/tmp`下，在没有`CMakeCahe.txt`的干扰下，`cmake -D`完全按照后面修改的配置信息**重新**编译OpenCV4！到`sudo make install` 结束后，可在新生成的`build/CMakeCache.txt`中查看`OPENCV_PYTHON_INSTALL_PATH_SETUPVARS:INTERNAL=lib/python3.6/site-packages`.

> INTERNAL表示的是相对(suffix)路径，在`cmake -D`中指定prefix为`/usr/local/`，所以opencv_python的安装路径为：`/usr/local/lib/python3.6/site-packages`.

#### symlink cv2.so into torch environment

```
$ ls /usr/local/lib/python3.6/site-packages/cv2/python-3.6/
cv2.cpython-36m-x86_64-linux-gnu.so [YES]
```

```
$ ln -s /usr/local/lib/python3.6/site-packages/cv2/python-3.6/cv2.cpython-36m-
x86_64-linux-gnu.so $ENV_PATH/lib/python3.6/site-packages/cv2.so
(torch) hxer@dislab:~/Code/opencv/build$ python
Python 3.6.2 |Continuum Analytics, Inc.| (default, Jul 20 2017, 13:51:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>>
```

## 总结

安装OpenCV4遇到很多坑，这段探索深刻明白"尽信书不如无书"！

参考的资料仅供参考，由于系统不同、版本不同甚至可能存在笔误导致参考信息不准确、不全面，在实践过程中需要专注、细致地查看提示信息，依据自己的直觉和程序运行的逻辑做出准确的判断；

## References:

- Buidling OpenCV with Conda on Linux;
- cmake -D中python参数参考：Building OpenCV for Anaconda Python 3、
- Installing opencv 3.1 with anaconda python3?-calocedrus回答